TRƯỜNG ĐẠI HỌC NHA TRANG

**KHOA SAU ĐẠI HỌC**



**BÁO CÁO**

**XÂY DỰNG CỔNG THÔNG TIN  
CÔNG TY 9THWONDER VIETNAM**

Giảng viên: TS. Phạm Thị Thu Thúy

Nghiên cứu sinh: Cao Viết Thắng

Khánh Hòa, tháng 07/2020

# MỤC LỤC

Đề mục Trang

Trang bìa i

[MỤC LỤC 2](#_Toc46779463)

[DANH SÁCH HÌNH VẼ 4](#_Toc46779464)

[DANH SÁCH BẢNG BIỂU 5](#_Toc46779465)

[MỞ ĐẦU 6](#_Toc46779466)

[CHƯƠNG 1. TỒNG QUAN 7](#_Toc46779467)

[1.1. Ontology 7](#_Toc46779468)

[1.1.1. Ontology là gì? 7](#_Toc46779469)

[1.1.2. Các cá thể (Individuals) - Thể hiện 8](#_Toc46779470)

[1.1.3. Các lớp (Classes) - Khái niệm 8](#_Toc46779471)

[1.1.4. Các thuộc tính (Properties) 8](#_Toc46779472)

[1.1.5. Các mối quan hệ (Relation) 9](#_Toc46779473)

[1.1.6. Ngôn ngữ OWL 9](#_Toc46779474)

[1.2. Giới thiệu về công ty 9thWonder Viet Nam 10](#_Toc46779475)

[CHƯƠNG 2. XÂY DỰNG ONTOLOGY 11](#_Toc46779476)

[2.1. Classes 11](#_Toc46779477)

[2.2. Object Properties 11](#_Toc46779478)

[2.3. Data Properties 12](#_Toc46779479)

[CHƯƠNG 3. XÂY DỰNG CỔNG THÔNG TIN 13](#_Toc46779480)

[3.1. Các công cụ sử dụng 13](#_Toc46779481)

[3.1.1. HTML, CSS, JS 13](#_Toc46779482)

[3.1.2. Python 13](#_Toc46779483)

[3.1.3. OwlReady2 15](#_Toc46779484)

[3.1.4. Django 16](#_Toc46779485)

[3.2. Kết quả đạt được 19](#_Toc46779486)

[CHƯƠNG 4. KẾT LUẬN 23](#_Toc46779487)

[TÀI LIỆU THAM KHẢO 24](#_Toc46779488)

# DANH SÁCH HÌNH VẼ

[Hình 1. Ontology 7](#_Toc46779489)

[Hình 2. Công ty 9thWonder 10](#_Toc46779490)

[Hình 3. Ứng dụng của Python 14](#_Toc46779491)

[Hình 4. Mô hình Model – Template – View 17](#_Toc46779492)

[Hình 5. MVT 18](#_Toc46779493)

[Hình 6. Logic của mô hình MVT 18](#_Toc46779494)

[Hình 7. Danh sách dự án 19](#_Toc46779495)

[Hình 8. Kết quả tìm kiếm Dự án 19](#_Toc46779496)

[Hình 9. Chi tiết dự án 19](#_Toc46779497)

[Hình 10. Danh sách Nhân viên 20](#_Toc46779498)

[Hình 11. Kết quả tìm kiếm Nhân viên 20](#_Toc46779499)

[Hình 12. Chi tiết Nhân viên 20](#_Toc46779500)

[Hình 13. Danh sách Chức vụ 21](#_Toc46779501)

[Hình 14. Danh sách Nhân viên có Chức vụ nhất định 21](#_Toc46779502)

[Hình 15. Danh sách Teams 22](#_Toc46779503)

[Hình 16. Danh sách Nhân viên thuộc Team nhất định 22](#_Toc46779504)

# DANH SÁCH BẢNG BIỂU

[Bảng 1. Object Properties 11](#_Toc46779505)

[Bảng 2. Data Properties 12](#_Toc46779506)

# MỞ ĐẦU

Ngày nay, sự bùng nổ tri thức cùng với các vấn đề giao lưu hội nhập quốc tế khiến mỗi chúng ta phải biết tận dụng những thành tựu của khoa học – công nghệ. Một trong những thành tựu nổi bật nhất của công nghệ thông tin (CNTT) là tạo ra website. Từ đó người dùng có thể truy cập lên website để làm nhiều việc như trao đổi, tìm kiếm thông tin.

Nhưng với sự đơn giản của Web hiện nay đã dẫn đến một số mặt hạn chế. Chẳng hạn khi muốn tìm kiếm một thông tin nào đó, chúng ta có thể dễ dàng bị lạc hay phải đối đầu với một lượng lớn thông tin không hợp lý và không liên quan được trả về từ kết quả tìm kiếm trên Web. Vì vậy, để khắc phục các yếu điểm của Web hiện tại khái niệm “Semantic Web” đã ra đời.

Semantic Web là một mạng lưới các thông tin được liên kết theo một cách thức để máy có thể dễ dàng xử lý thông tin ở mức độ toàn cầu. Chúng ta có thể hiểu đơn giản về nó như là một cách trình bày dữ liệu có hiệu quả trên mạng toàn cầu, hoặc là một cơ sở dữ liệu được kết nối toàn cầu. Kỹ thuật Web hiện tại chỉ cho phép chúng ta lưu trữ thông tin và tìm kiếm thông tin trên những thông tin đã được lưu trữ. Chính vì vậy mà sự ra đời của Semantic Web là một bước tiến vượt bậc so với kỹ thuật Web hiện tại dựa vào khả năng làm việc với thông tin của chúng thay vì chỉ đơn thuần là lưu trữ thông tin. Với sự lớn mạnh và khả năng lưu trữ thông tin ngữ nghĩa, Semantic Web sẽ trở thành một thế hệ Web cho tương lai, thế hệ Web mà chúng ta đang chờ đợi.

Qua môn học Công nghệ Web ngữ nghĩa, em sẽ trình bày về cách để tạo ra được một trang web cổng thông tin của công ty 9thWonder VietNam, tìm kiếm dự án, thông tin nhân viên của công ty dựa trên cơ sở là Ontology.

# TỒNG QUAN

## Ontology

### Ontology là gì?

Trong khoa học máy tính, một ontology là một mô hình dữ liệu biểu diễn một lĩnh vực và được sử dụng để suy luận về các đối tượng trong lĩnh vực đó và mối quan hệ giữa chúng. Ontology cung cấp một bộ từ vựng chung bao gồm các khái niệm, các thuộc tính quan trọng và các định nghĩa về các khái niệm và các thuộc tính này. Ngoài bộ từ vựng, ontology còn cung cấp các ràng buộc, đôi khi các ràng buộc này được coi như các giả định cơ sở về ý nghĩa mong muốn của bộ từ vựng, nó được sử dụng trong một miền mà có thể được giao tiếp giữa người và các hệ thống ứng dụng phân tán hỗn tạp khác.



Hình 1. Ontology

Các ontology được sử dụng như là một biểu mẫu trình bày tri thức về thế giới hay một phần của nó. Các ontology thường miêu tả gồm:

* Các cá thể: Các đối tượng cơ bản, nền tảng
* Các lớp: Các tập hợp, hay kiểu của các đối tượng
* Các thuộc tính: Thuộc tính, tính năng, đặc điểm, tính cách, hay các thông số mà các đối tượng có và có thể đem ra chia sẻ.
* Các mối liên hệ: Các con đường mà các đối tượng có thể liên hệ tới một đối tượng khác.

Bộ từ vựng ontology được xây dựng trên cơ sở tầng RDF và RDFS, cung cấp khả năng biểu diễn ngữ nghĩa mềm dẻo cho tài nguyên Web và có khả năng hỗ trợ lập luận.

### Các cá thể (Individuals) - Thể hiện

*Các cá thể* là các thành phần cơ bản, nền tảng của một ontology. Các cá thể trong một ontology có thể bao gồm các đối tượng cụ thể như con người, động vật, cái bàn… cũng như các cá thể trừu tượng như các thành viên hay các từ. Một ontology có thể không cần bất kỳ một cá thể nào, nhưng một trong những lý do chính của một ontology là để cung cấp một ngữ nghĩa của việc phân lớp các cá thể, mặc dù các cá thể này không thực sự là một phần của ontology.

### Các lớp (Classes) - Khái niệm

*Các lớp* là các nhóm, tập hợp các đối tượng trừu tượng. Chúng có thể chứa các cá thể, các lớp khác, hay là sự phối hợp của cả hai. Các ontology biến đổi tuỳ thuộc vào cấu trúc và nội dung của nó: Một lớp có thể chứa các lớp con, có thể là một lớp tổng quan (chứa tất cả mọi thứ), có thể là lớp chỉ chứa những cá thể riêng lẻ, Một lớp có thể xếp gộp vào hoặc bị xếp gộp vào bởi các lớp khác. Mối quan hệ xếp gộp này được sử dụng để tạo ra một cấu trúc có thứ bậc các lớp, thường là với một lớp thông dụng nhất kiểu Thing ở trên đỉnh và các lớp rất rõ ràng kiểu 2002, Ford ở phía dưới cùng.

### Các thuộc tính (Properties)

*Các đối tượng* trong ontology có thể được mô tả thông qua việc khai báo các thuộc tính của chúng. Mỗi một thuộc tính đều có tên và giá trị của thuộc tính đó. Các thuộc tính được sử dụng để lưu trữ các thông tin mà đối tượng có thể có. Ví dụ, đối với một cá nhân có thể có các thuộc tính: Họ\_tên, ngày\_sinh, quê\_quán, số\_cmnd…

Giá trị của một thuộc tính có thể có các kiểu dữ liệu phức tạp.

### Các mối quan hệ (Relation)

Một trong những ứng dụng quan trọng của việc sử dụng các thuộc tính là để mô tả mối liên hệ giữa các đối tượng trong ontology*. Một mối quan hệ* là một thuộc tính có giá trị là một đối tượng nào đó trong ontology.

Một kiểu quan hệ quan trọng là kiểu quan hệ xếp gộp (subsumption). Kiểu quan hệ này mô tả các đối tượng nào là các thành viên của các lớp nào của các đối tượng. Hiện tại, việc kết hợp các ontology là một tiến trình được làm phần lớn là thủ công, do vậy rất tốn thời gian và đắt đỏ. Việc sử dụng các ontology là cơ sở để cung cấp một định nghĩa thông dụng của các thuật ngữ cốt lõi có thể làm cho tiến trình này trở nên dễ quản lý hơn. Hiện đang có các nghiên cứu dựa trên các kỹ thuật sản sinh để nối kết các ontology, tuy nhiên lĩnh vực này mới chỉ hiện hữu về mặt lý thuyết.

### Ngôn ngữ OWL

*OWL (The Web Ontology Language)* là một ngôn ngữ gần như XML dùng để mô tả các hệ cơ sở tri thức. OWL là một ngôn ngữ đánh dấu dùng để xuất bản và chia sẻ dữ liệu trên Internet thông qua những mô hình dữ liệu gọi là “ontology”. Ontology mô tả một lĩnh vực (domain) và diễn tả những đối tượng trong lĩnh vực đó cùng những mối quan hệ giữa các đối tượng này. OWL là phần mở rộng về từ vựng của RDF và được kế thừa từ ngôn ngữ DAML+OIL Web ontology – một dự án được hỗ trợ bởi W3C. OWL biểu diễn ý nghĩa của các thuật ngữ trong các từ vựng và mối liên hệ giữa các thuật ngữ này để đảm bảo phù hợp với quá trình xử lý bởi các phần mềm.

OWL được xem như là một kỹ thuật trọng yếu để cài đặt cho Semantic Web trong tương lai.

OWL được thiết kế đặc biệt để cung cấp một cách thức thông dụng trong việc xử lý nội dung thông tin của Web. Ngôn ngữ này được kỳ vọng rằng sẽ cho phép các hệ thống máy tính có thể đọc được thay thế cho con người. Vì OWL được viết bởi XML, các thông tin OWL có thể dễ dàng trao đổi giữa các kiểu hệ thống máy tính khác nhau, sử dụng các hệ điều hành và các ngôn ngữ ứng dụng khác nhau.

Mục đích chính của OWL là sẽ cung cấp các chuẩn để tạo ra một nền tảng để quản lý tài sản, tích hợp mức doanh nghiệp và để chia sẻ cũng như tái sử dụng dữ liệu trên Web. OWL được phát triển bởi nó có nhiều tiện lợi để biểu diễn ý nghĩa và ngữ nghĩa hơn so với XML, RDF và RDFS, và vì OWL ra đời sau các ngôn ngữ này, nó có khả năng biểu diễn các nội dung mà máy có thể biểu diễn được trên Web.

## Giới thiệu về công ty 9thWonder Viet Nam



Hình 2. Công ty 9thWonder

9thWonder VietNam là một chi nhánh của công ty 9thWonder chuyên về làm việc về các lĩnh vực công nghệ thông tin.

Hiện tại công ty đang làm các dự án công nghệ thông tin như Website (App – Landing…. ) và Mobile App. Như P360, App Âm nhạc, Mazi

Chia thành 4 team (group):

* Group 1 – Tiger Beer
* Group 2 – Corona Beer
* Group 3 – Heiniken Beer
* Group Mazi

Công ty được quản lý bởi anh Đỗ Khắc Duy.

* Các Project Manager – anh Châu, anh Đức
* Các Leader – anh Danh, anh Lợi, anh Liêm
* Các Developer – Thắng, An, Ý,…
* Các Tester – chị Trang, chị Dương,…

Website chính thức của công ty:

<https://www.9thwonder.com/>

# XÂY DỰNG ONTOLOGY

## Classes

DU\_AN: lớp Dự án

CHUC\_VU: lớp Chức vụ

TEAM: lớp Team

NHAN\_VIEN: lớp Nhân viên

## Object Properties

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Name | Domains | Ranges | Invert Of | Miêu tả |
| 1 | Duan\_tao\_ra\_boi\_team | DUAN | TEAM | Team\_tao\_ra\_duan |  |
| 2 | Team\_ta\_ra\_duan |  |  | Duan\_tao\_ra\_boi\_team |  |
| 3 | Nv\_thuoc\_team | NHAN\_VIEN | TEAM | Team\_co\_nv |  |
| 4 | Team\_co\_nv |  |  | Nv\_thuoc\_team |  |
| 5 | Nv\_co\_cv | NHAN\_VIEN | CHUC\_VU | Cv\_cua\_nv |  |
| 6 | Cv\_cua\_nv |  |  | Nv\_co\_cv |  |

Bảng 1. Object Properties

## Data Properties

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Name | Domains | SubProerty Of | Type | Miêu tả |
| 1 | DA | DUAN |  |  |  |
| 2 | Ngon\_ngu | DUAN | DA | String |  |
| 3 | Ten\_DA | DUAN | DA | String |  |
| 4 | Tom\_tat | DUAN | DA | String |  |
| 5 | Data\_nhan\_vien | NHAN\_VIEN |  |  |  |
| 6 | Avatar | NHAN\_VIEN | Data\_nhan\_vien | String |  |
| 7 | Dia\_chi | NHAN\_VIEN | Data\_nhan\_vien | String |  |
| 8 | Email | NHAN\_VIEN | Data\_nhan\_vien | String |  |
| 9 | Ho\_ten | NHAN\_VIEN | Data\_nhan\_vien | String |  |
| 10 | Ngay\_sinh | NHAN\_VIEN | Data\_nhan\_vien | DateTime |  |
| 11 | Noi\_sinh | NHAN\_VIEN | Data\_nhan\_vien | String |  |
| 12 | Que\_quan | NHAN\_VIEN | Data\_nhan\_vien | String |  |
| 13 | SDT | NHAN\_VIEN | Data\_nhan\_vien | String |  |
| 14 | Ten\_cv | CHUC\_VU |  | String |  |
| 15 | Ten\_team | TEAM |  | String |  |

Bảng 2. Data Properties

# XÂY DỰNG CỔNG THÔNG TIN

## Các công cụ sử dụng

### HTML, CSS, JS

Như bao trang web khác, thành phần cơ bản của một trang web bao giờ cũng sẽ có 3 thành phần: HTML, CSS, JS và tất nhiên, trang web của em cũng sẽ được tạo ra từ 3 ngôn ngữ này.

* **HTML**cung cấp *cấu trúc cơ bản* của các trang web, được cải tiến và sửa đổi bởi các công nghệ khác như CSS và JavaScript.
* **CSS**được sử dụng để kiểm soát *trình bày, định dạng và bố cục*.
* **JavaScript**được sử dụng để kiểm soát *hành vi* của các yếu tố khác nhau.

### Python

Python là một ngôn ngữ lập trình cấp cao và đang được sử dụng rộng khắp cộng đồng các developer.

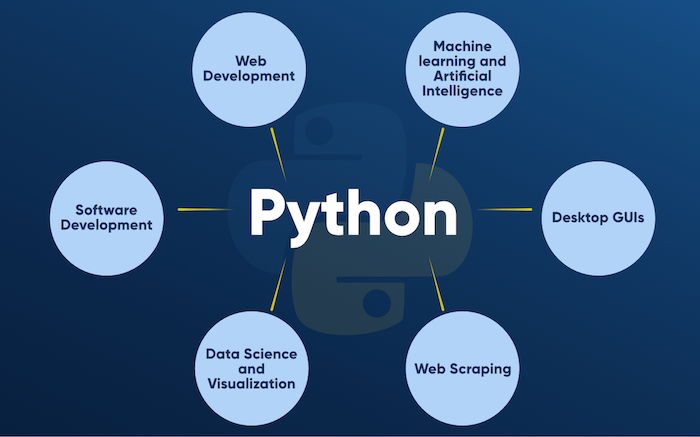
Python đã được phát triển chủ yếu tập trung vào khả năng đọc code. Cú pháp của nó cho phép lập trình viên có thể lập trình với ít dòng code hơn. Python là một ngôn ngữ lập trình cho phép các developer có thể làm việc nhanh chóng và có thể tích hợp các hệ thống hiệu quả hơn.

* **Các đặc điểm quan trọng**

Python có nhiều lý do để trở nên phổ biến và có nhiều nhu cầu sử dụng. Dưới đây là một vài lý do:

* Tập trung vào khả năng đọc code, các đoạn code ngắn hơn, và dễ viết hơn.
* Các lập trình viên có thể thể hiện các ý tưởng logic với ít dòng code hơn so với các ngôn ngữ khác như C++ và Java.
* Python hỗ trợ đa mẫu hình lập trình giống như lập trình hướng đối tượng, lập trình hàm và mệnh lệnh, hoặc lập trình thủ tục.
* Nó cung cấp các thư viện hỗ trợ mở rộng (Django để phát triển web, Pandas cho phân tích dữ liệu, OwlReady2 để tương tác với Ontology, v.v.).
* Ngôn ngữ kiểu động (Kiểu dữ liệu dựa trên giá trị được gán)
* Triết lý là “Tinh giản là tốt nhất”.
* **Lĩnh vực ứng dụng của Python**

Python đang là một ngôn ngữ rất hot trong thời điềm hiện tại ví tính ưu việt, đa dụng của nó. Python hiện đang được ứng dụng vào trong rất nhiều lĩnh vực: Website, Apps, Data,…



Hình 3. Ứng dụng của Python

### OwlReady2

OwlReady2 là một module để tương tác với Ontology trong Python.

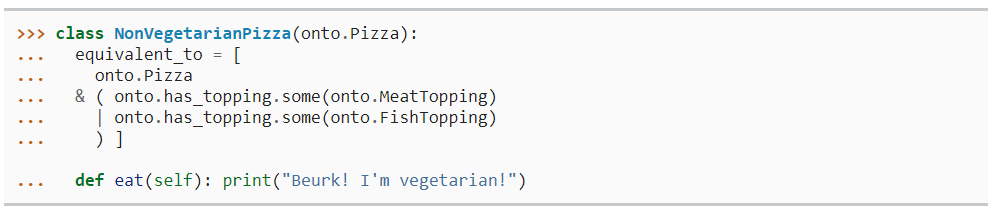
OwlReady 2 có thể load các file .owl để tạo thành các đối tượng trong Python, qua đó sửa đổi và lưu trữ lại hay thực hiện query (searching).

Vài ví dụ sử dụng OwlReady2:

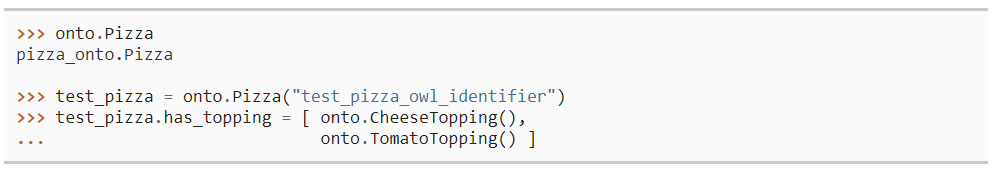
* **Load Ontology từ file .owl ở Local hay Internet:**



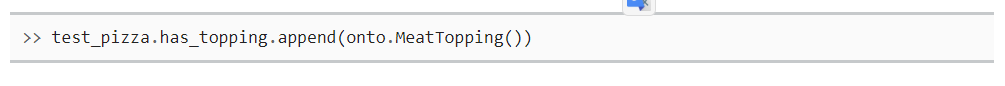
* **Tạo mới Classes trên Ontology, kết hợp OWL với các phương thức của Python:**



* **Truy cập Classes và tạo ra các Individuals / Instances:**



* **Trong OwlReady2, ta có thể liệt kê hầu hết các danh sách và thực hiện việc sửa đổi, thêm/xóa các item từ danh sách. OwlReady2 sẽ tự động cập nhật bộ tứ RDF.**



* **Xuất ra file OWL**



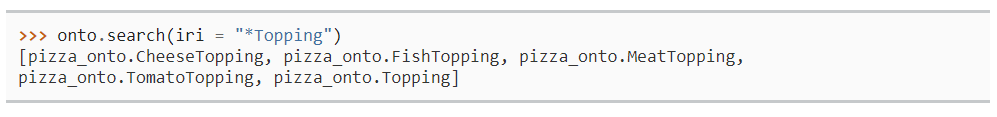
* **Truy vấn trên file OWL bằng OwlReady2**

OwlReady2 cung cấp một phương thức query cơ bản là .search(), với các tham số truyền vào các keyword như:

* **IRI**: tìm kiếm các thực thể bằng full IRI.
* **type**: tìm kiếm thưc thể của một Classes nhất định.
* **subclass\_of**: tìm kiếm các SubClass của một Classes nhất định
* **is\_a:** tìm kiếm cả thực thể và SubClass của một Classes nhất định
* **any object**: dữ liệu hoặc tên thuộc tính

Giá trị được liên kết với mỗi từ khóa có thể là một giá trị đơn hoặc danh sách giá trị. Dấu “\*” có thể được sử dụng như một jocker trong các giá trị chuỗi.

Ví dụ: để tìm kiếm tất cả các thực thể có IRI kết thúc bằng ‘Topping”:



Ngoài ra, dấu “\*” có thể được sử dụng như một ký tự đại diện cho bất kỳ đối tượng nào. Ví dụ: để tìm kiếm tất cả các thực thể có liên quan đến một thực thể khác có quan hệ “has\_topping”.



Khi muốn kết quả trả về là một giá trị duy nhất, ta dùng phương thức .search\_one(), hoạt động tương tự như phương thức .search():



Một số phương thức tìm kiếm phức tạp hơn như search bằng SQPARQL. Ta sẽ dùng WORLDS của OwlReady2, tham khảo thêm tại:

<https://pythonhosted.org/Owlready2/world.html>

### Django

#### Django là gì?

Django là một trong những Web Framework phổ biển nhất được viết bằng Python, cung cấp nhiều tính năng cho việc phát triển web về bảo mật, data access, sesion, routing,…

Django sử dụng mô hình MVT (Model – View – Template). Về phương thức hoạt động sẽ giống với mô hình MVC, trong đó V (View) sẽ tương đương với C (Controller), T (Template) sẽ tương đương với V (View) ở các framework khác.

Django được sáng lập bởi Adrian Holovaty và Simon\_Willison vào năm 2003 và publish phiên bản đầu tiên vào năm 2005. Hiện nay, tại thời điểm bài viết này, Django đã release đến phiên bản 3.0.8.

Cộng đồng sử dụng và maintaince Django là khá lớn. Nếu ta tìm kiếm với từ khóa Django tại trang Stackoverflow thì sẽ có khoảng 240.000 kết quả, còn trên Github là 322.000. Github của Django có khoảng ~22000 Fork và 28000 commit.

Một số website lớn sử dụng Django có thể kể đến như: Instagram, Mozilla, Disqus, National Geographic, Bitbucker….

#### Kiến trúc Django

Kiến trúc Django gồm ba phần chính:

1. Tập hợp các công cụ giúp người lập trình thao tác với database dễ dàng hơn.
2. Phần định nghĩa các mẫu văn bản (vd như nội dung mail, text thông báo nào đó, v.v…).
3. Phần framework, xử lý luồng chạy giữa thao thác của user với database.

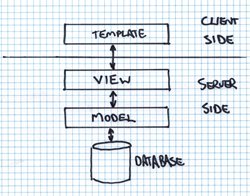
Dưới đây là sơ đồ mô tả phần 1 với 2



Hình 4. Mô hình Model – Template – View

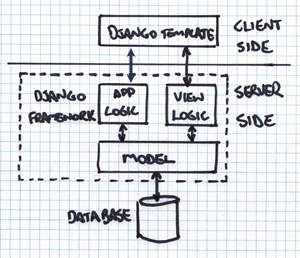
* Model là các tools làm việc với data và database.
* Template là các văn bản giao diện mà bạn sẽ nhận từ các designers.
* Còn View, nó có phải là một controller không.

Nhìn vào mô hình của Django, ta có thể thấy phần nào khá giống với những mô hình framework ngoài thị trường với cấu trúc MVC (Model - View - Controller). Nhưng với Django thì có tên khác là MTV (Model - Template - View). Lúc này View của Django sẽ đóng vai trò như controller của mô hình MVC, chỉ là cách gọi tên khác nhau thôi.



Hình 5. MVT

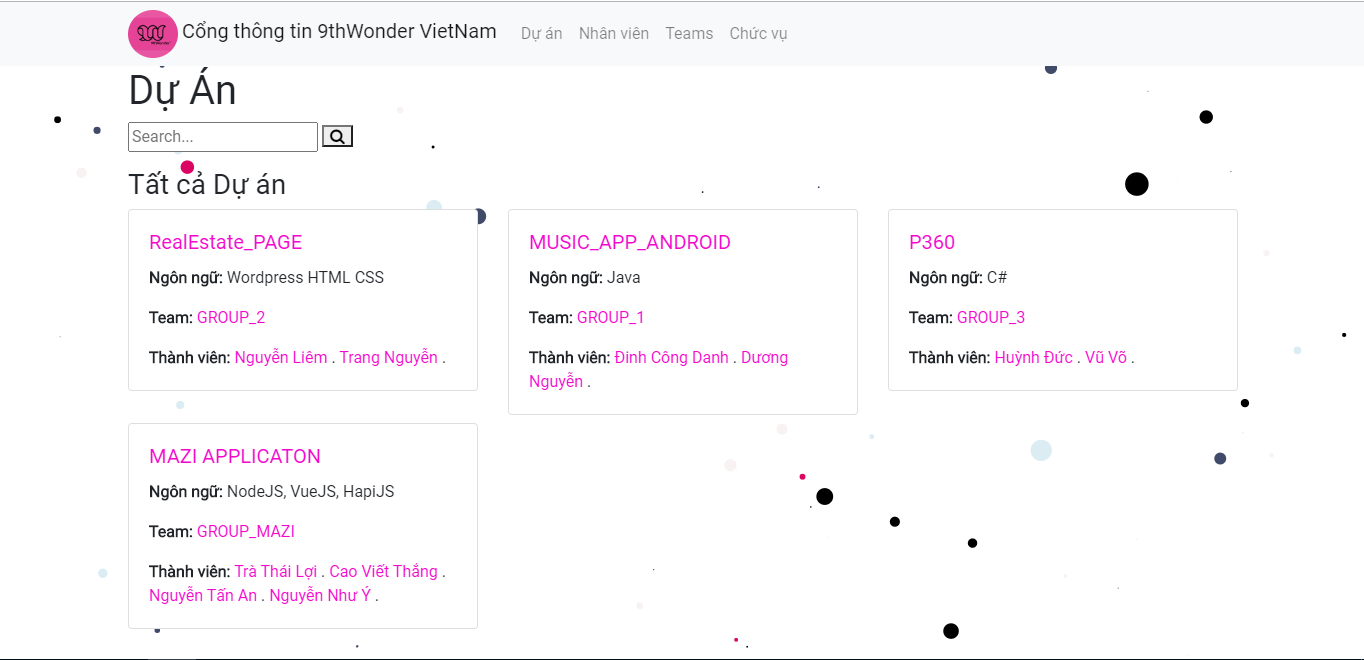
Cụ thể trong view sẽ là những gì.



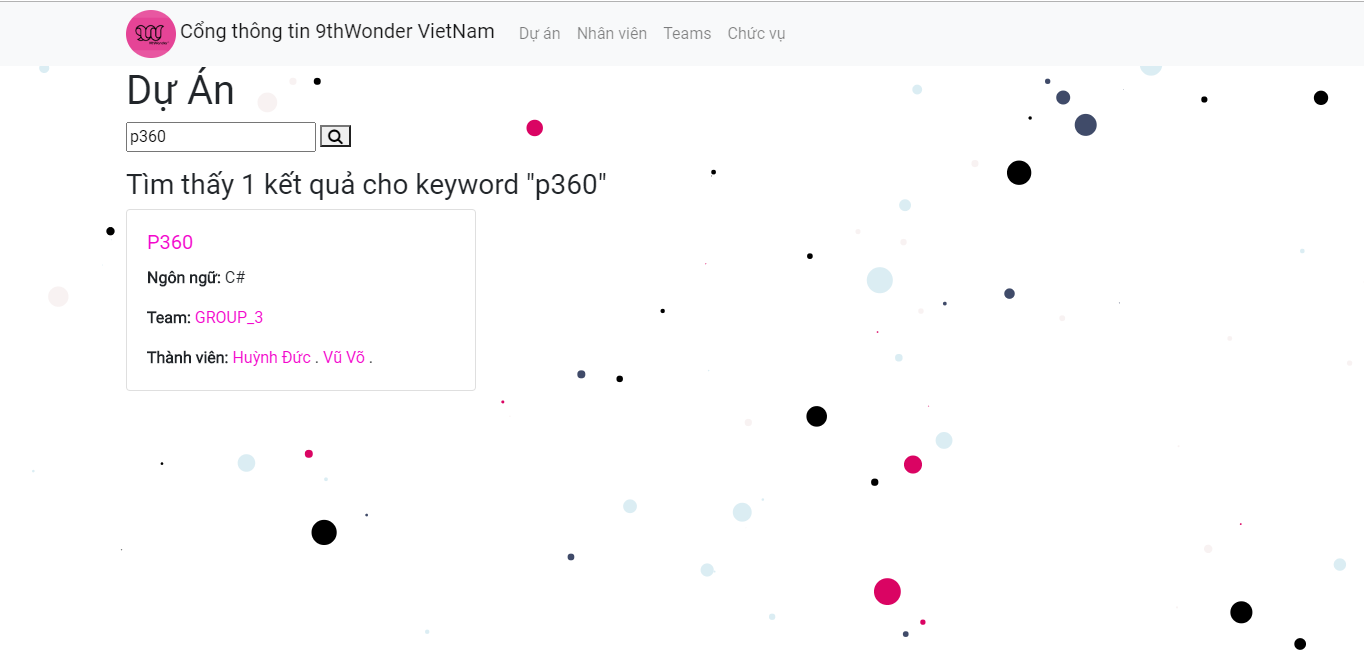
Hình 6. Logic của mô hình MVT

## Kết quả đạt được

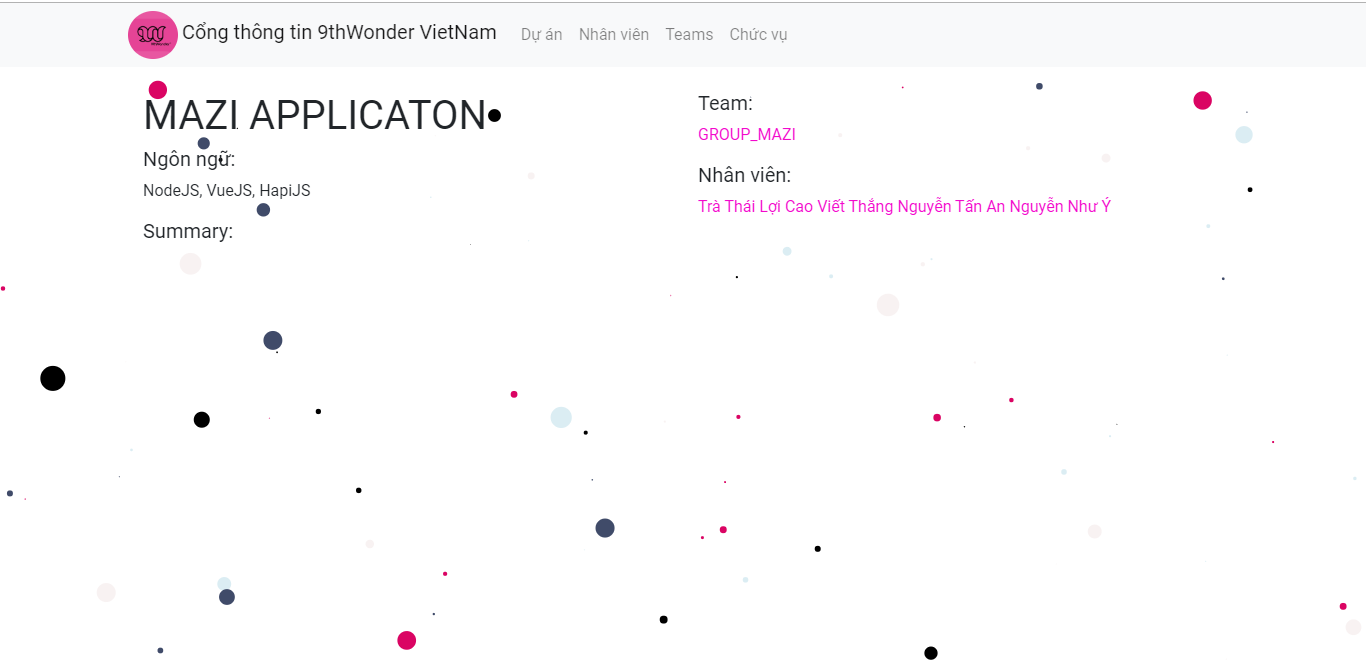
* **Dự án:**



Hình 7. Danh sách dự án

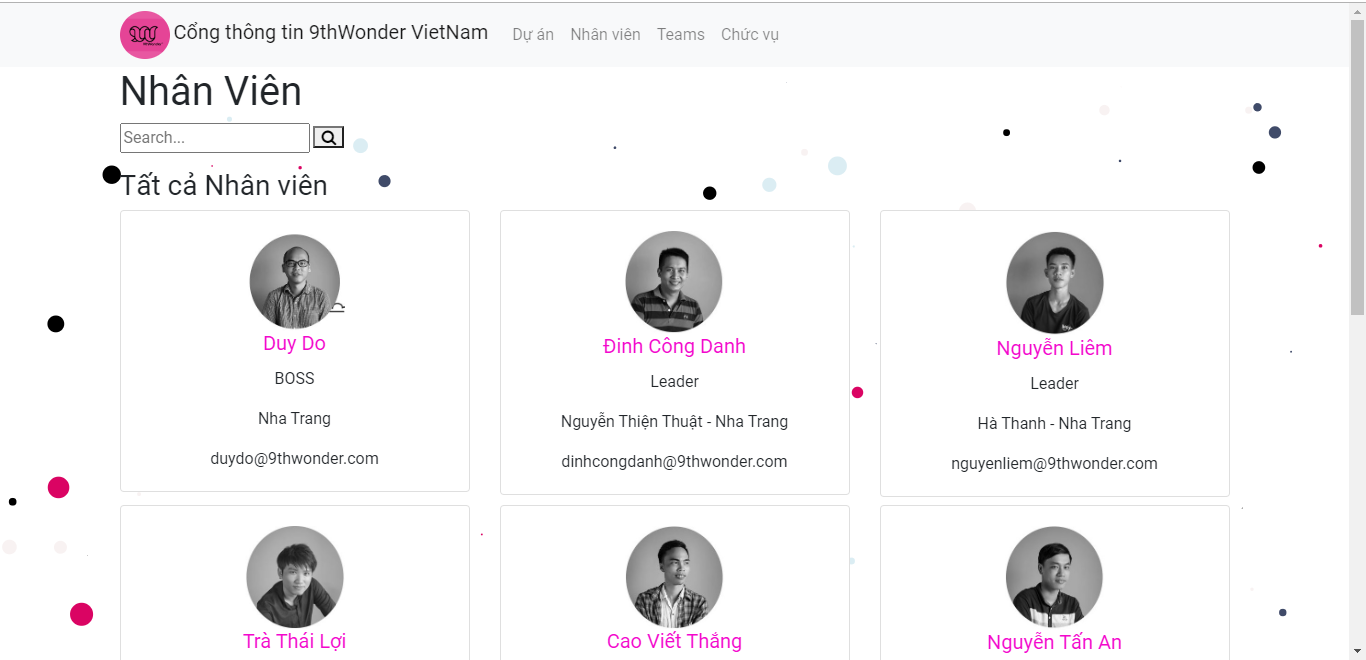


Hình 8. Kết quả tìm kiếm Dự án

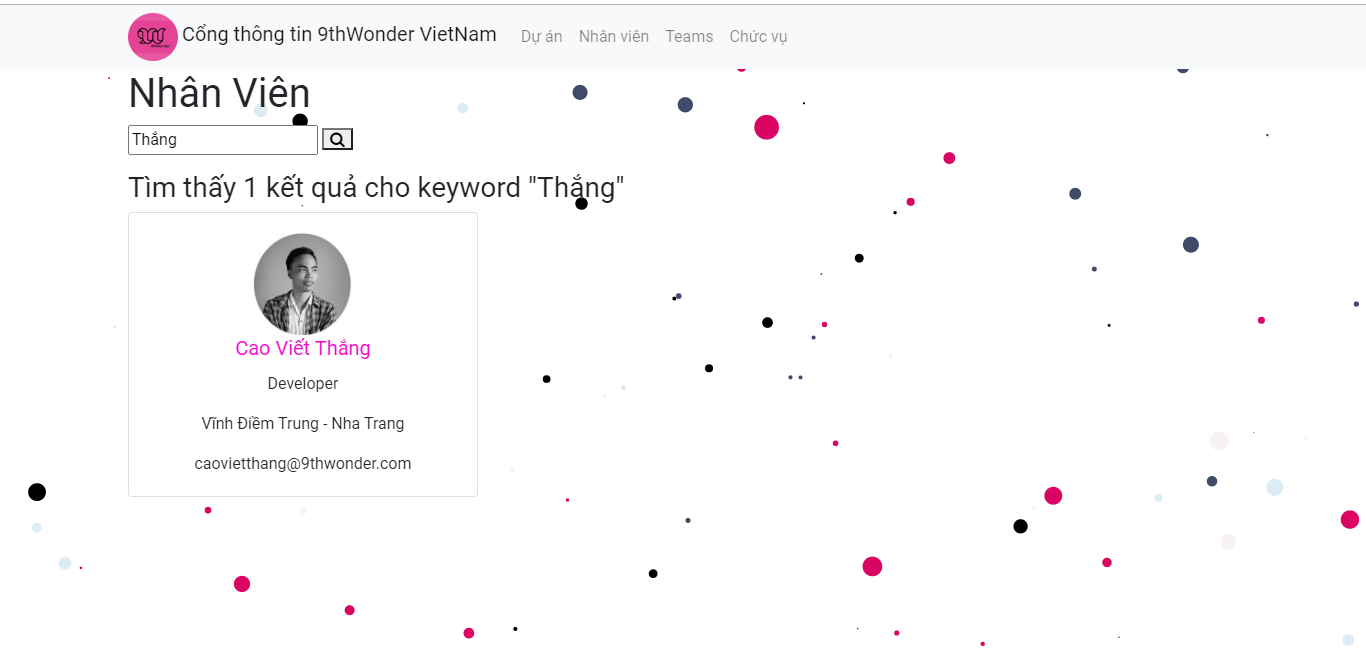


Hình 9. Chi tiết dự án

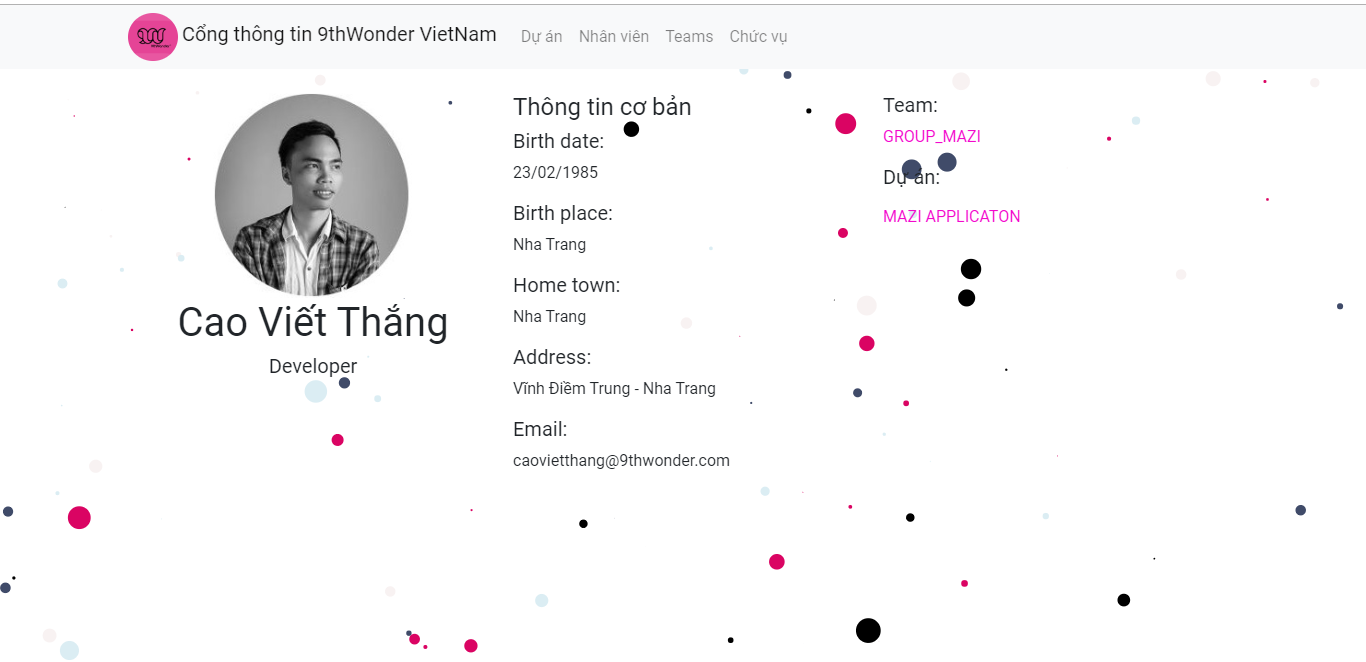
* **Nhân viên:**



Hình 10. Danh sách Nhân viên

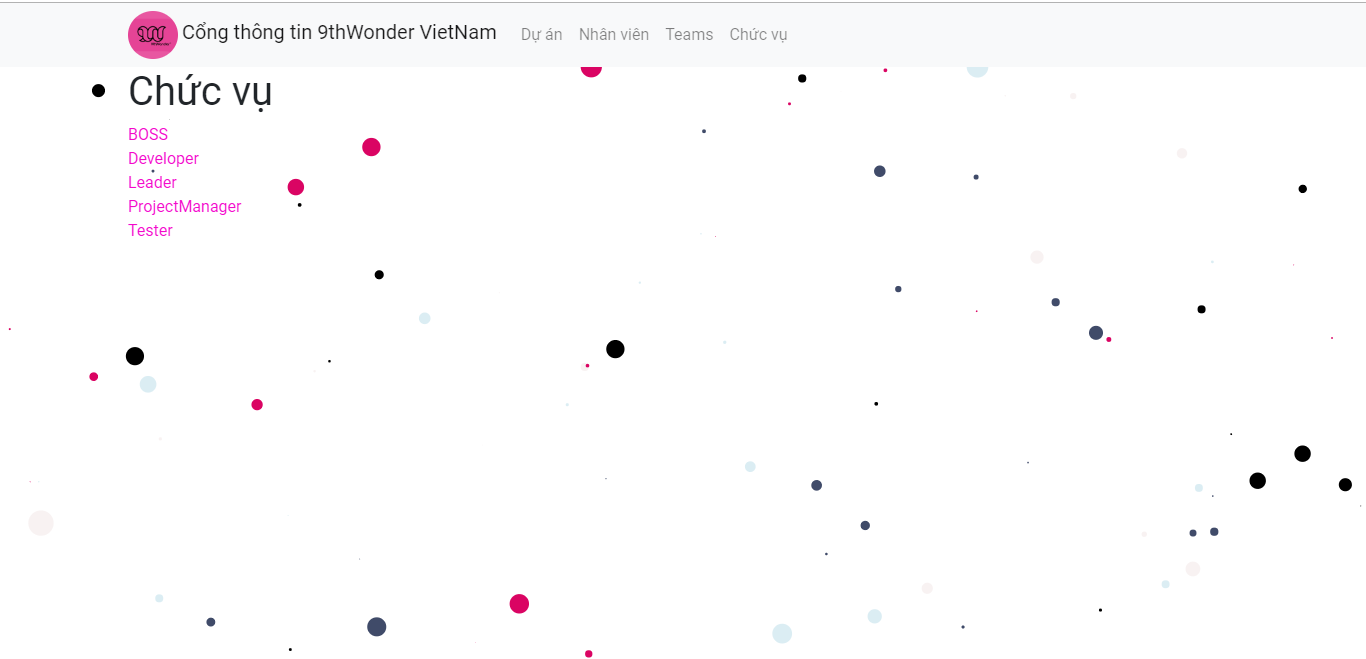


Hình 11. Kết quả tìm kiếm Nhân viên

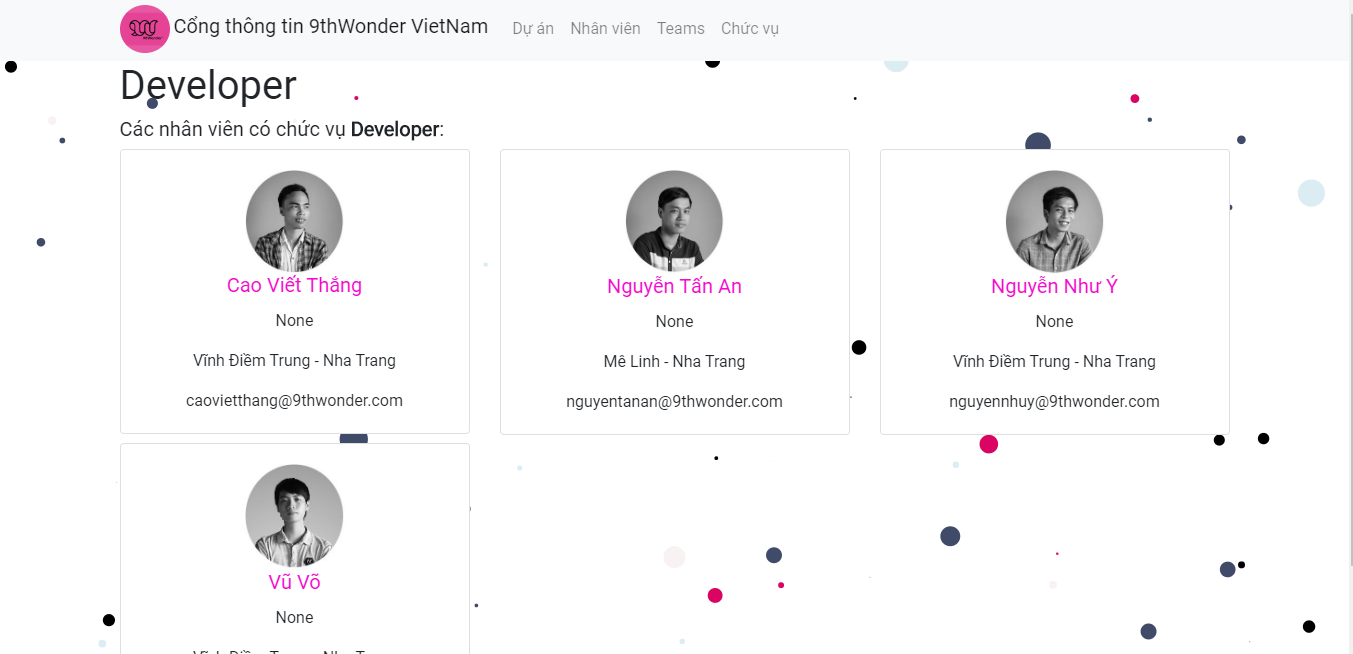


Hình 12. Chi tiết Nhân viên

* **Chức vụ:**

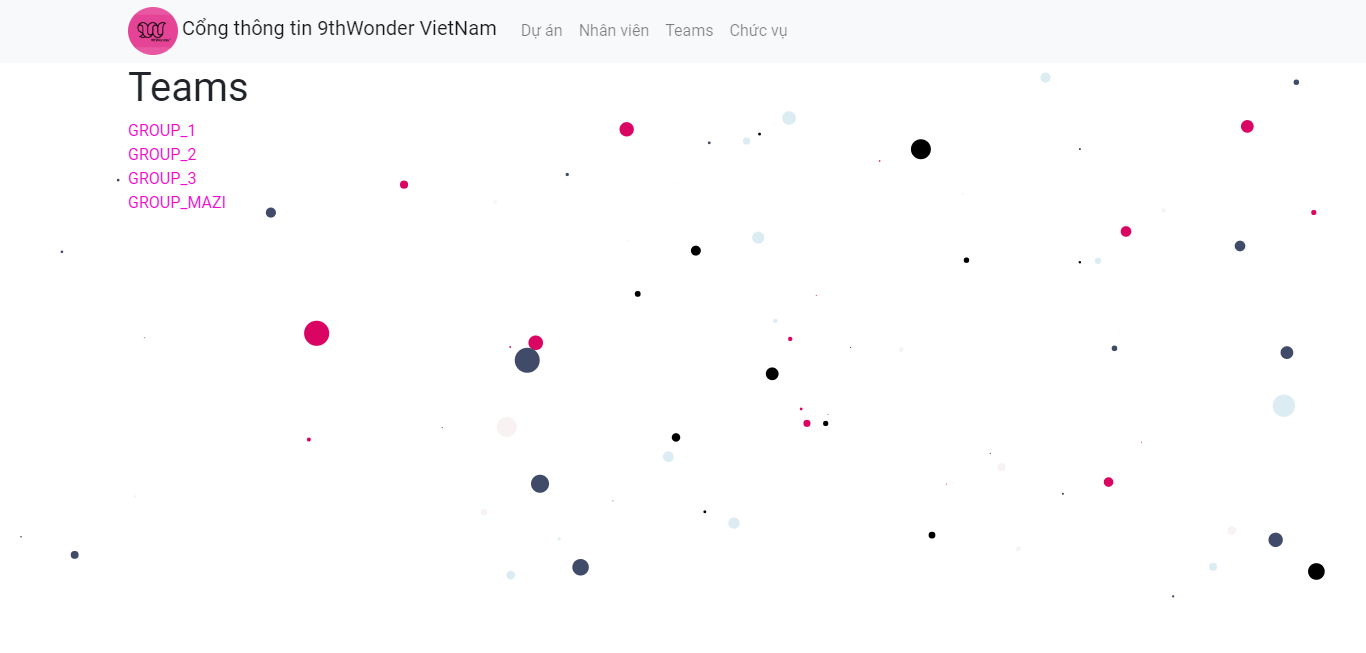


Hình 13. Danh sách Chức vụ

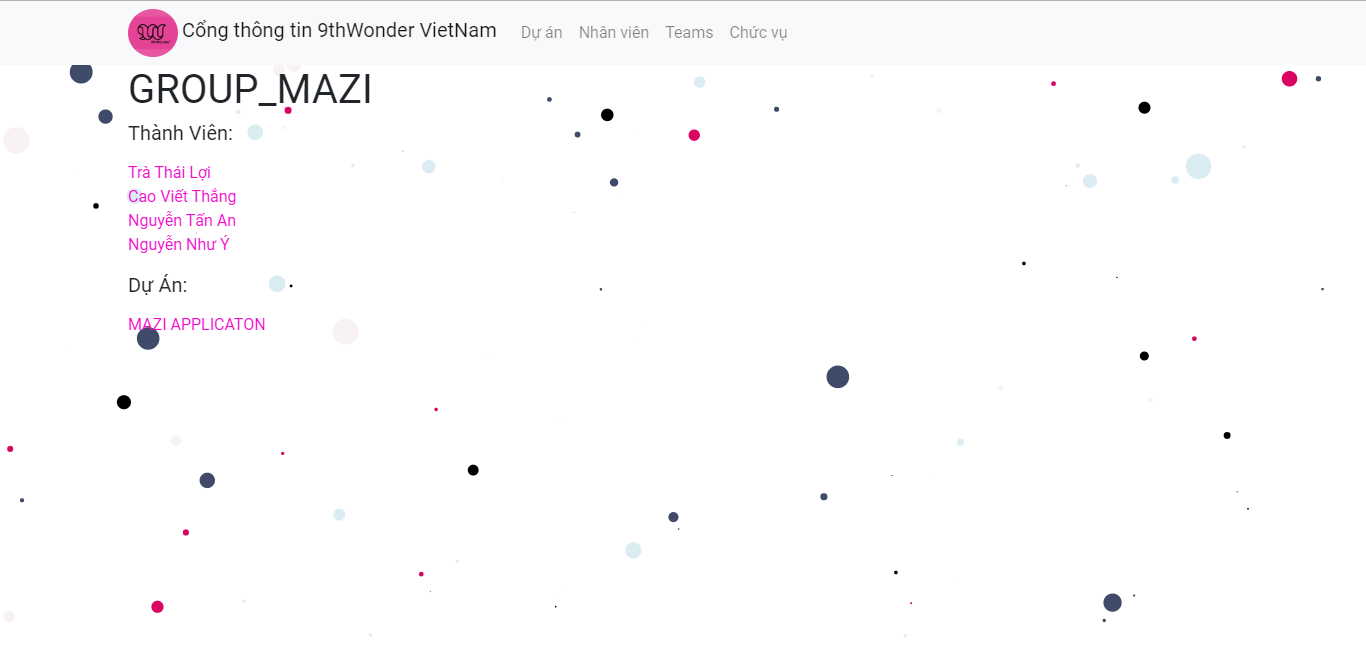


Hình 14. Danh sách Nhân viên có Chức vụ nhất định

* **Teams:**



Hình 15. Danh sách Teams



Hình 16. Danh sách Nhân viên thuộc Team nhất định

# KẾT LUẬN

Sau thời gian nghiên cứu, em đã tìm hiểu được tổng quan về Ontology và xây dựng được cổng thông tin công ty 9thWonder Viet Nam. Bài báo cáo có thể còn nhiều thiếu sót, mong thầy có thể bỏ qua và giúp nhóm hoàn thiện hơn.

# TÀI LIỆU THAM KHẢO

**Tiếng Việt:**

* Website:

1. Vietnam Lab Center Co., Ltd, “ Django - Demo với website hiển thị giá Crypto - Phần 1”, <https://blog.vietnamlab.vn/2019/10/09/django-demo-voi-website-hien-thi-gia-crypto/>
2. TechTalk, Giới thiệu về Python, <https://techtalk.vn/gioi-thieu-ve-python.html>

**Tiếng Anh:**

* Website:

3. Django Documents, <https://docs.djangoproject.com/en/3.0/>

4. OwlReady2 Documents, <https://pythonhosted.org/Owlready2/>